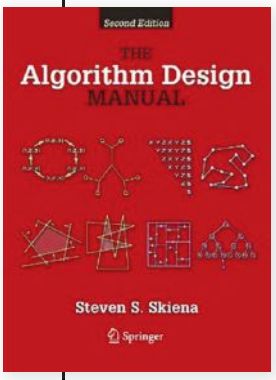
SEE TEXT ONLY

BOOKreview

by Joseph Kong

The Algorithm Design Manual

2nd Edition by Steven S. Skiena

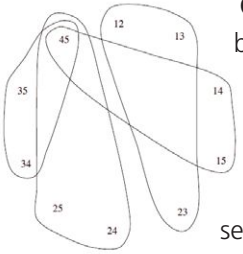


Publisher .....*Springer; 2nd edition (July 2008)*
Print List Price *$89.95*
eBook List Price.....*$69.99*
ISBN-10.....*1848000693*
ISBN-13.....*978-1848000698*
Pages *730*

This is one of the best books I've read in the past year. It's a steady tour through the world of algorithms with just the right amount of handholding (definitely not for novice programmers though).

The Algorithm Design Manual is divided into two parts. The first part, Chapters 1–10, describes the fundamentals of algorithm design, including data structures, dynamic programming, backtracking, modeling, and so on. Each chapter also contains several "war stories," which detail Skiena's experience with real-world problems. These stories are quite entertaining and help lighten the mood on some rather dense material.

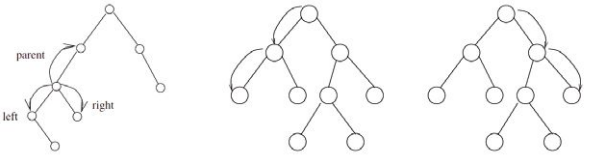
The second part, Chapters 11–19, is a catalog of algorithmic resources (for example, data structures, implementations, and so on) and is intended for browsing and reference. The idea is that instead of solving algorithmic problems from scratch, you use an existing one as a starting point (or the solution).



Chapter 1, Introduction to Algorithm Design, begins by describing what an algorithm is before going into detail about modeling, which is the art of breaking down a problem into one or more well-understood problems (whose solution can be found in the second part of this book).

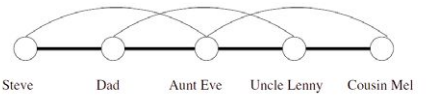
Chapter 2, Algorithm Analysis, describes algorithm efficiency and introduces the "big Oh" notation. Here, I found Skiena's explanation of "big Oh" to be quite good—definitely better than a lot of other texts.

Chapter 3, Data Structures, describes arrays, linked lists, stacks, queues, dictionaries, binary trees, priority queues, and hash tables all in the context of algorithm design. I found this chapter to be a very good refresher course.



Chapter 4, Sorting and Searching, stresses how sorting can be applied to solve other problems. The concepts of binary search and divide-and-conquer are also described. Several algorithms are detailed here, including heapsort, mergesort, quicksort, and distribution sort.

Chapter 5, Graph Traversal, begins by describing the different types of graphs (for example, directed, undirected, weighted, unweighted, and so on) and data structures used for graphs (that is, adjacency matrixes and adjacency lists) before describing graph traversal techniques (that is, breadth-first search and depth-first search).

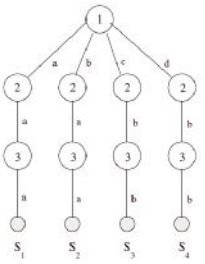


Chapter 6, Weighted Graph Algorithms, is a continuation of Chapter 5, which delves into weighted graphs, including discussions about spanning trees. Here, I found the war story about typing text via a telephone—the old-school way, where inputting 2 meant either the letter A, B, or C—and how to identify the correct letter via context to be rather entertaining and informative.

Chapter 7, Combinatorial Search and Heuristic Methods, describes how to systematically iterate through all the possible solutions (or configura-

tions) of a problem (or search space). The concepts of backtracking and search pruning are described here. I particularly enjoyed Skiena's walk-through on how to solve any Sudoku problem, as it really highlighted the power of backtracking and search pruning.

Chapter 8, Dynamic Programming, as the title suggests, describes dynamic programming. I found Skiena's explanation of the subject to be quite clear and easy to understand.



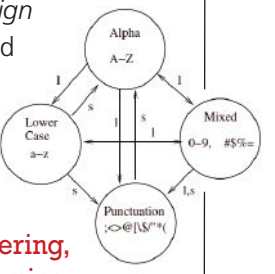
Chapter 9, Intractable Problems and Approximation Algorithms, introduces techniques for proving that no efficient algorithm exists for a particular problem. The idea is that if you can prove that searching for an efficient algorithm is pointless, you can focus your efforts elsewhere.

Chapter 10, How to Design Algorithms, is simply a four-page checklist to keep in mind when designing algorithms. While not much of a chapter, I greatly appreciated this list.

Chapters 11–19 contain a catalog of algorithmic resources detailing data structures, numerical problems, combinatorial problems, graph problems, computational geometry, and set and string problems. This portion of the book is affectionately known as "The Hitchhiker's Guide to Algorithms."

If I had to say something negative about this book, it'd be that it's dry and dense in parts—it's definitely not lazy Sunday afternoon reading. But these are minor nits.

The bottom line is that T*he Algorithm Design Manual* is a solid book and if you haven't read it yet, you should. ●



JOSEPH KONG is a self-taught computer enthusiast who dabbles in the fields of exploit development, reverse code engineering, rootkit development, and systems programming (FreeBSD, Linux, and Windows). He is the author of the critically acclaimed *Designing BSD Rootkits* and *FreeBSD Device Drivers*. For more information about Joseph Kong visit www.thestackframe.org or follow him on Twitter @JosephJKong.